Autonomous Quadcopter Drone Race

Aneri Hiren Desai <anerid@uci.edu>, Derek Tran <derekt5@uci.edu>. Isaiah Cabugos <icabugos@uci.edu>
Jasera Abdurrashid <jabdurra@uci.edu> Rohankumar Barouliya <rbarouli@uci.edu>
Guided by: Prof. Yasser Shoukry
Professional Master of Embedded and Cyber-physical Systems Program, University of California, Irvine

www.mecps.uci.edu

Abstract

This project develops a fully autonomous mobility stack for mid-sized quadrotor drones designed for high-speed racing. Using COEX Clover drones equipped with onboard sensors and an Arducam 100 fps mono global shutter camera, the system enables vision-based navigation through multiple race checkpoints. Implemented in the Robot Operating System (ROS), the autonomy stack integrates localization, perception, path planning, and control. Initially validated using a ViCON motion tracking system, the setup now achieves fully onboard operation using ArUco marker-based localization and YOLO-based checkpoint detection.

The multi-modal control layer combines a Model Predictive Controller (MPC) for precise trajectory tracking and a Reinforcement Learning (RL) model for agile decision-making. The RL racing policy, trained for 140 million steps across lightweight simulation and Gazebo, achieves average speeds of 2.85 m/s with peaks of 4.16 m/s. A neural network-trained MPC is converted to TensorFlow Lite and deployed on a Raspberry Pi 4 for real-time execution. The result is an end-to-end, vision-driven, learning-enabled drone platform capable of fast, agile, and fully autonomous flight.

Objectives

Build a Fully Autonomous Stack
- Integrat localization, perception, planning, and control within ROS for mid-sized quadrotors

Enable Onboard Vision
- Deploy ArUco marker-based localization and YOLO checkpoint detection—no external tracking

Develop Dual Control Modes
- Combine MPC for stable trajectory tracking with RL for aggressive racing maneuvers

Deploy on Embedded Hardware
- Convert neural network MPC to TensorFlow Lite for real-time execution on Raspberry Pi 4

Validate Through Sim-to-Real Pipeline
- Train in lightweight simulation → fine-tuned in Gazebo → tested on physical hardware

Achieve High-Speed Autonomous Flight
- Demonstrate vision-driven racing with onboard compute at competitive speeds

Materials and Methods

Hardware

- COEX Clover Drone 4.2
- Raspberry Pi 4 Model B
- PTK Votik 9497 Servo
- Arducam 100fps Mono Global Shutter USB Camera
- XYG- Raspberry Pi R Camera
- Power Distribution Board
- MR30 connector (for motor speed control)
- Gemfan Drone racing gates

Software

- ROS Noetic

Simulation/Model Acquisition
- Gazebo version 11.15.1
- Python 3.9
- Tensorflow version 2.13.0
- TF-agents version 0.17.0
- Numpy version 1.24.3
- Nvidia CUDA Development Kit 11.8
- Gymnasium version 0.29.1

Diagrams/Figures/Experiments

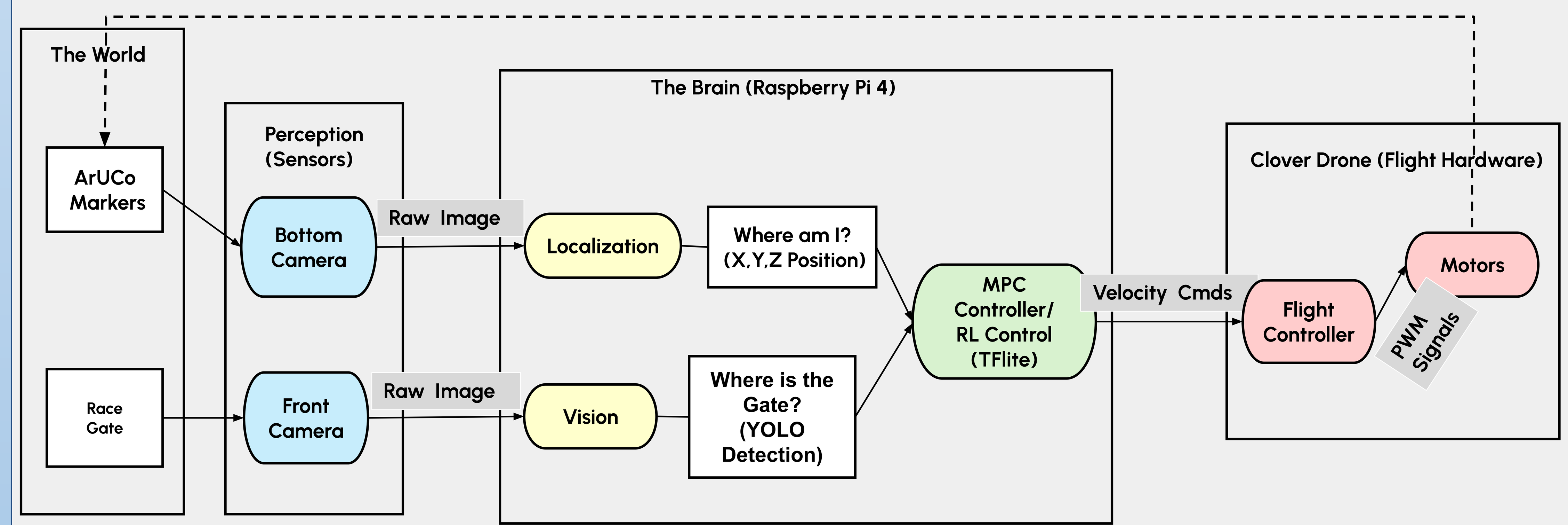


Fig 1: System Architecture

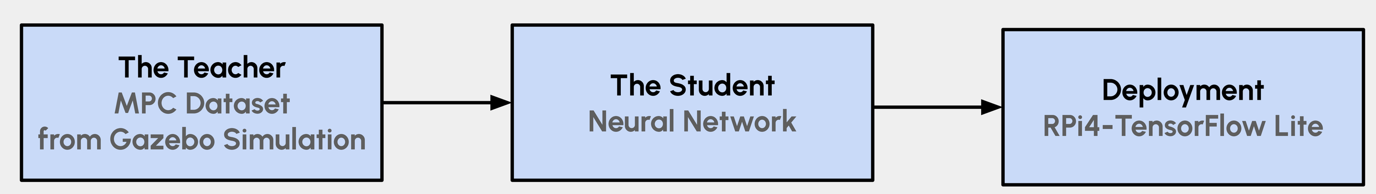


Fig 2: MPC Deployment Strategy

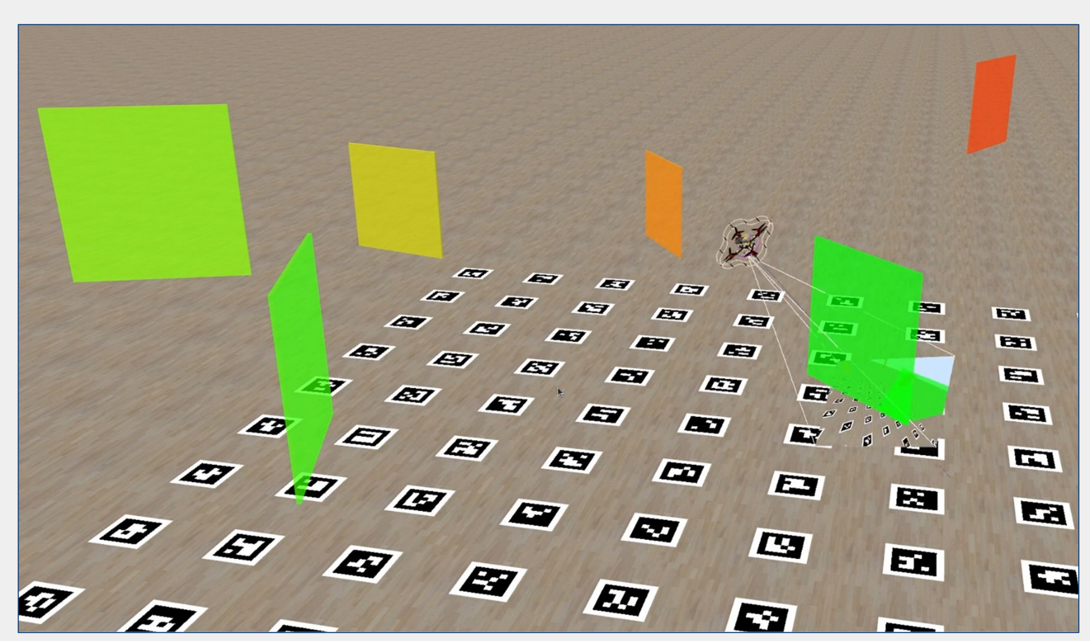


Fig 3(a): Drone Flying in Gazebo Simulator

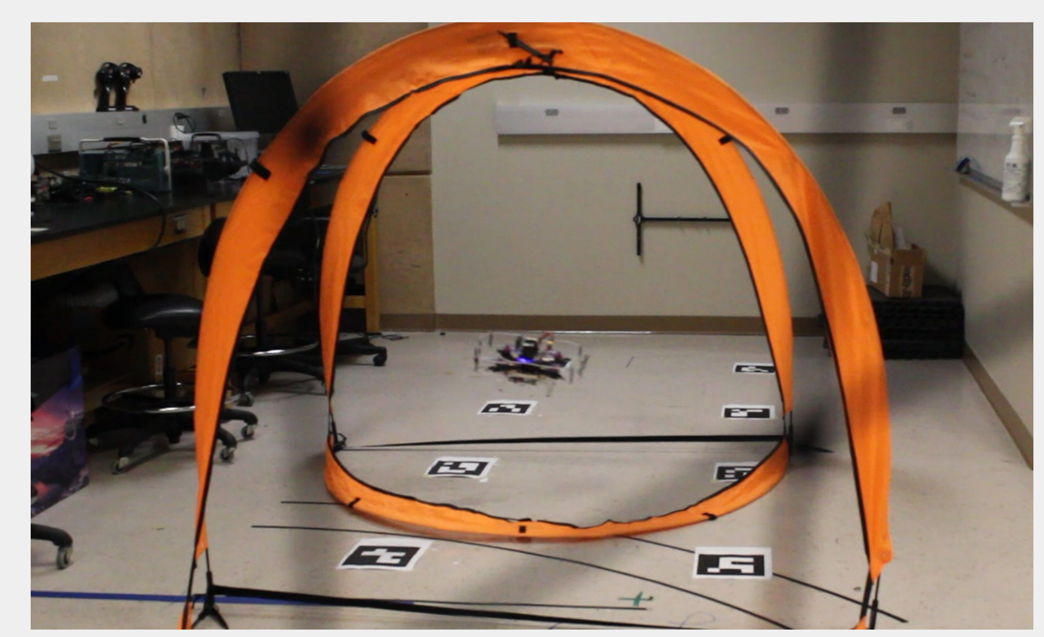


Fig 3(b): Drone Flying between Gates in Real World

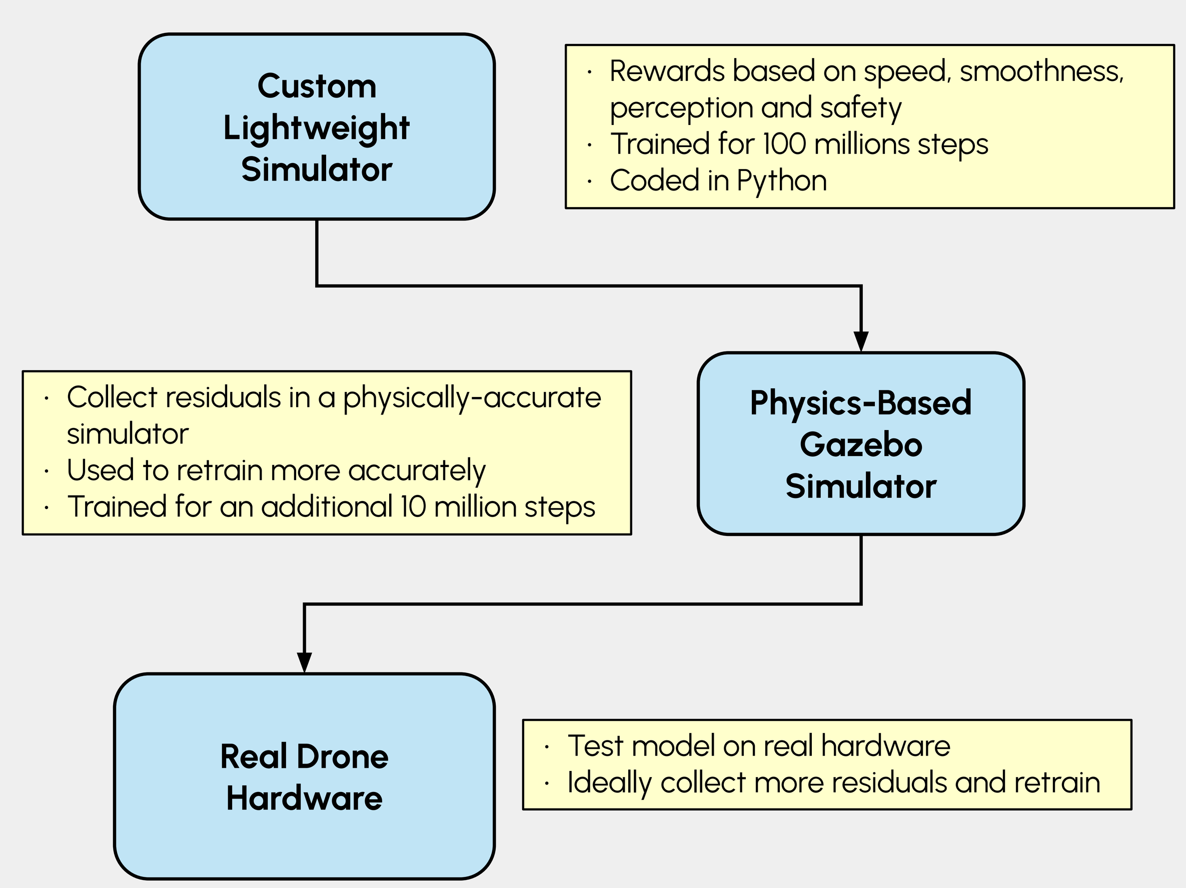


Fig 4(a): RL Simulation to Real Pipeline

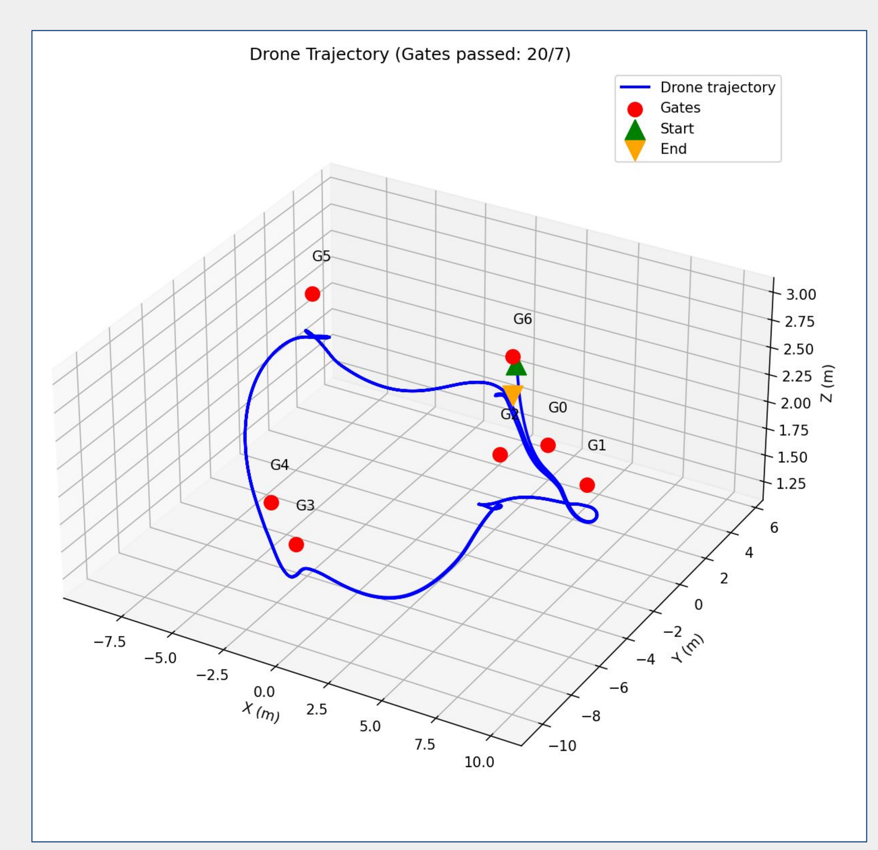


Fig 4(b): RL Racing Model Visualization

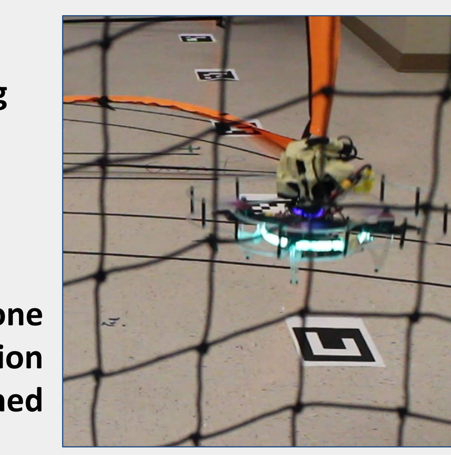
Results

Model Predictive Control: Experimental Results
- Validated core MPC logic by implementing a Z-axis (altitude) controller in Gazebo — performed as expected.
- Expanded to a full 3D MPC, achieving stable and accurate flight along linear trajectory with minor deviations.
- During sharp turns (e.g., 90° corners), the drone failed to execute precise maneuvers, often cutting corners or missing gates.

RL Racing Model Control: Experimental Results
- Racing model trained for 140 Million Steps in lightweight custom sim, able to lap racetrack multiple times per 1500 steps
- Achieves average speed of 2.85m/s and top speed of 4.16m/s

Fig 5a (Left): Perception Reporting Gate Location

Fig 5b (Right): Drone Flying with Perception Hardware Attached



YOLO-based perception model : Experimental Results
- Validated Gates/Object detection using YOLO in real-time with high accuracy.
- Successfully calculating accurate distance from Arducam camera to Gate using pixel calculation method, and further generating target co-ordinates i.e. gate coordinates, assuming camera as epi-center.

Additional Information

- This project is supervised by **Prof. Yasser Shoukry** at the **Resilient Cyber-Physical Systems Lab**, UC Irvine

Team Member	Key Contribution
Aneri Hiren Desai	MPC Development, MPC Neural Network Model Development, Testing and Training of NN model
Derek Tran	Drone Management and AI Camera
Isaiah Cabugos	RL Development, Localization calibration, Flight Control Configuration and Testing
Jasera Abdurrashid	MPC training, ROS Simulation and Testing
Rohankumar Barouliya	Camera integration, YOLO pipeline, Target distance estimation

References

- COEX Clover: https://clover.coex.tech/en/
- AirPilot: Interpretable PPO-based DRL Auto-Tuned Nonlinear PID Drone Controller for Robust Autonomous Flights : https://arxiv.org/abs/2404.00204
- Actor-Critic Model Predictive Control: Differentiable Optimization meets Reinforcement Learning: https://rpg.ifi.uzh.ch/research_drone_racing.html
- You Only Look Once: Unified, Real-Time Object Detection: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf
- Champion-level drone racing using deep reinforcement learning: https://www.nature.com/articles/s41586-023-06419-4